A Transition from RDF to Petri Nets

Jan Paredaens

Universiteit Antwerpen

11.11.11

RDF

Constraints in RDF

WikiPedia

Petri Nets

Decidability - Axioms

Conclusion

RDF

Ramanathan V. Guha
1965 -
1997 RDF - Netscape Corp.

RDF

RDF (Resource Description Framework) is a W3C recommendation for publishing structured data on the web.

RDF databases arrange information in simple triples consisting of a subject, a predicate and an object.

```
exstaff:85740          exterms:addressUSA          _:johnaddress .
exstaff:85741          exterms:addressUSA          _:anaddress .
_:johnaddress          exterms:street              "1501 Grant Avenue" .
_:johnaddress          exterms:city                "Bedford" .
_:johnaddress          exterms:state               "Massachusetts" .
_:johnaddress          exterms:postalCode          "01730" .
_:exterms:addressUSA   exterms:country             "USA" .
_:exterms:addressNL    exterms:country             "NL" .
```

RDF

An RDF-Graph is a finite set of triples (s, p, o).

RDF

An RDF-Graph is a finite set of triples (s, p, o).

As such it is one big ternary relation.

RDF

An RDF-Graph is a finite set of triples (s, p, o).

As such it is one big ternary relation.

Semantically we cannot use the relational model.

RDF

An RDF-Graph is a finite set of triples (s, p, o).

As such it is one big ternary relation.

Semantically we cannot use the relational model.

For constraints we cannot use keys, functional dependencies, referential integrity, ...

RDF

An RDF-Graph is a finite set of triples (s, p, o).

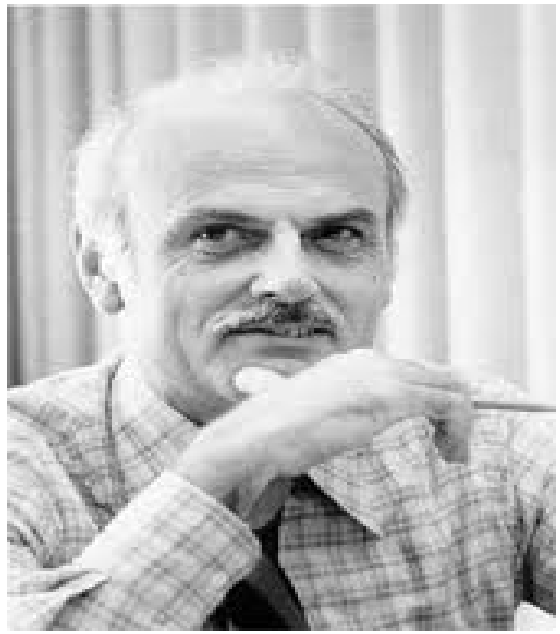
As such it is one big ternary relation.

Semantically we cannot use the relational model.

For constraints we cannot use keys, functional dependencies, referential integrity, ...

We have to use patterns and embeddings.

Constraints in RDF



Ted Codd
1923 - 2003
1970 Relational DB

Constraints in RDF

Constraints in RDF are mainly used for:

- Consistency checking.
- Consistent query answering.
- Data cleaning.
- Semantic query optimization.
- Distributed data management.

Constraints in RDF

```
exstaff:85740          exterms:addressUSA          _:johnaddress .
exstaff:85741          exterms:addressUSA          _:anaddress .
_:johnaddress          exterms:street              "1501 Grant Avenue" .
_:johnaddress          exterms:city                "Bedford" .
_:johnaddress          exterms:state               "Massachusetts" .
_:johnaddress          exterms:postalCode          "01730" .
_:exterms:addressUSA   exterms:country             "USA" .
_:exterms:addressNL    exterms:country             "NL" .
```

Triple Generating Constraint

For every address in the USA we need a state.

$$TGC = \{(\$x, \texttt{exterms:addressUSA}, \$y)\}, \{(\$y, \texttt{exterms:state}, \$z)\}$$
$$\{(\$x, \texttt{exterms:addressUSA}, \$y)\}, \{(\$y, \texttt{exterms:state}, \$z)\}$$

Constraints in RDF

```
exstaff:85740          exterms:addressUSA          _:johnaddress .
exstaff:85741          exterms:addressUSA          _:anaddress .
_:johnaddress          exterms:street              "1501 Grant Avenue" .
_:johnaddress          exterms:city                "Bedford" .
_:johnaddress          exterms:state               "Massachusetts" .
_:johnaddress          exterms:postalCode          "01730" .
_:exterms:addressUSA   exterms:country             "USA" .
_:exterms:addressNL    exterms:country             "NL" .
```

Functional Constraint

Every address in the USA has at most one state.

$FC = \{(\$x, \texttt{exterms:addressUSA}, \$y), (\$y, \texttt{exterms:state}, \$z)\}, y \rightarrow z$

Constraints in RDF

```
exstaff:85740          exterms:addressUSA          _:johnaddress .
exstaff:85741          exterms:addressUSA          _:anaddress .
_:johnaddress          exterms:street              "1501 Grant Avenue" .
_:johnaddress          exterms:city                "Bedford" .
_:johnaddress          exterms:state               "Massachusetts" .
_:johnaddress          exterms:postalCode          "01730" .
_:exterms:addressUSA   exterms:country             "USA" .
_:exterms:addressNL    exterms:country             "NL" .
```

Forbidding Constraint

Every address in the Netherlands has no state.

$$FBC = \{(\$x, \texttt{exterms:addressNL}, \$y)\}, \{(\$y, \texttt{exterms:state}, \$z)\}$$

Constraints in RDF

```
exstaff:85740          exterms:addressUSA          _:johnaddress .
exstaff:85741          exterms:addressUSA          _:anaddress .
_:johnaddress          exterms:street              "1501 Grant Avenue" .
_:johnaddress          exterms:city                "Bedford" .
_:johnaddress          exterms:state               "Massachusetts" .
_:johnaddress          exterms:postalCode          "01730" .
_:exterms:addressUSA   exterms:country             "USA" .
_:exterms:addressNL    exterms:country             "NL" .
```

(in)Equality Generating Constraint

For every address in the USA the city and the state are different

$$EGC = \{(\$x, \texttt{exterms:addressUSA}, \$y)\}, \{(\$y, \texttt{exterms:state}, \$z), (\$y, \texttt{exterms:city}, \$z')\}, \$z \neq \$z'$$

WikiPedia

Jimmy Wales
1966 -
2003 Wikipedia

WikiPedia

DBpedia is an open-access, large RDF repository that stores encyclopedic information retrieved from Wikipedia, and other URLs. It contains 1 billion triples.

WikiPedia

Every person in DBpedia should have at most one birth date.

$$\{(\$person, dbo : birthDate, \$y),$$
$$(\$person, dbo : birthDate, \$z)\},$$
$$\{(\$y = \$z)\}\}.$$

```
Ask {?person dbo:birthDate ?y.
     ?person dbo:birthDate ?z.
      FILTER(?y != ?z)}
```

As the total number of people with a recorded birth date is 465498, we conclude that 0.72% of all people on DBpedia violate this constraint!

WikiPedia

The next constraint expresses that no person is born after his/her death:

```
Ask {?person dbo:birthDate ?birth .
     ?person dbo:deathDate ?death .
      FILTER (?death<?birth)}
```

This constraint is violated 150 times.

WikiPedia

When checking the constraint that no extinct bird still live we get a violation:

```
Ask{?animal dbpprop:extinct ?z.
    ?animal dbpprop:classis :Bird.
    ?animal dbpprop:status ?y
    FILTER (?y != "EX"@en )}
```

This constraint is not satisfied by the Antillean Cave Rail (Nesotrochis debooyi).

Petri Nets

Carl Adam Petri
1926 - 2010
1962 Petri Net

Petri Nets

We represent a Petri Net ¡¡TEKENING¿¿

by an RDF Graph:

```
$p_1$  $t$  $p3$
```

Petri Nets

$$TGC_1 = \{(\$p_1, \$t, \$p_3), (\$p_2, \$t, \$p_4)\}, \{(\$p_1, \$t, \$p_4)\}$$

Petri Nets

$$TGC_1 = \{(\$p_1, \$t, \$p_3), (\$p_2, \$t, \$p_4)\}, \{(\$p_1, \$t, \$p_4)\}$$

$$TGC_2 = \{(\$p_1, \$t, \$p_2)\}, \{(\$p_1, Place, \$p_1), (\$p_2, Place, \$p_2),$$

Petri Nets

$$TGC_1 = \{(\$p_1, \$t, \$p_3), (\$p_2, \$t, \$p_4)\}, \{(\$p_1, \$t, \$p_4)\}$$

$$TGC_2 = \{(\$p_1, \$t, \$p_2)\}, \{(\$p_1, Place, \$p_1), (\$p_2, Place, \$p_2),$$

$$FB_1 = \{(\$p_1, \$t, \$p_2), (\$t, Place, \$t)\}$$

Petri Nets

$$TGC_1 = \{(\$p_1, \$t, \$p_3), (\$p_2, \$t, \$p_4)\}, \{(\$p_1, \$t, \$p_4)\}$$

$$TGC_2 = \{(\$p_1, \$t, \$p_2)\}, \{(\$p_1, Place, \$p_1), (\$p_2, Place, \$p_2),$$

$$FB_1 = \{(\$p_1, \$t, \$p_2), (\$t, Place, \$t)\}$$

Connected.

Petri Nets

Two problems ¡¡TEKENING¿¿

$TGC_1 = \{(\$p, Input, \$t)\}, \{(\$p, Place, \$p)\}$

Petri Nets

Two problems ¡¡TEKENING¿¿

$$TGC_1 = \{(\$p, Input, \$t)\}, \{(\$p, Place, \$p)\}$$

$$TGC_2 = \{(\$t, Output, \$p)\}, \{(\$p, Place, \$p)\}$$

Petri Nets

Two problems ¡¡TEKENING¿¿

$$TGC_1 = \{(\$p, Input, \$t)\}, \{(\$p, Place, \$p)\}$$

$$TGC_2 = \{(\$t, Output, \$p)\}, \{(\$p, Place, \$p)\}$$

$$FBC_1 = \{(\$p, Input, \$t), (\$t, Place, \$t)\}$$

Petri Nets

Two problems ¡¡TEKENING¿¿

$$TGC_1 = \{(\$p, Input, \$t)\}, \{(\$p, Place, \$p)\}$$

$$TGC_2 = \{(\$t, Output, \$p)\}, \{(\$p, Place, \$p)\}$$

$$FBC_1 = \{(\$p, Input, \$t), (\$t, Place, \$t)\}$$

$$FBC_2 = \{(\$t, Output, \$p), (\$t, Place, \$t)\}$$

Petri Nets

Two problems ¡¡TEKENING¿¿

$TGC_1 = \{(\$p, Input, \$t)\}, \{(\$p, Place, \$p)\}$

$TGC_2 = \{(\$t, Output, \$p)\}, \{(\$p, Place, \$p)\}$

$FBC_1 = \{(\$p, Input, \$t), (\$t, Place, \$t)\}$

$FBC_2 = \{(\$t, Output, \$p), (\$t, Place, \$t)\}$

$EGC = \{(\$x_1, \$x_2, \$x_3)\}, \$x_2 = Input \ \vee \ \$x_2 = Output \ \vee \ \$x_2 = Place$

Petri Nets

Two problems ¡¡TEKENING¿¿

$TGC_1 = \{(\$p, Input, \$t)\}, \{(\$p, Place, \$p)\}$

$TGC_2 = \{(\$t, Output, \$p)\}, \{(\$p, Place, \$p)\}$

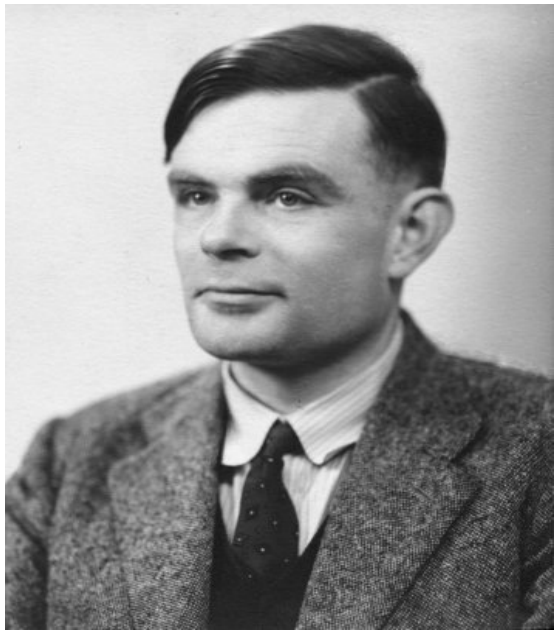
$FBC_1 = \{(\$p, Input, \$t), (\$t, Place, \$t)\}$

$FBC_2 = \{(\$t, Output, \$p), (\$t, Place, \$t)\}$

$EGC = \{(\$x_1, \$x_2, \$x_3)\}, \$x_2 = Input \ \vee \ \$x_2 = Output \ \vee \ \$x_2 = Place$

Connected.

Decidability - Axioms



Alan Turing
1912 - 1954
1936 Turing Machine

Decidability - Axioms

- More constraints on RDF;

Decidability - Axioms

- More constraints on RDF;
- Also a recursive constraint;

Decidability - Axioms

- More constraints on RDF;
- Also a recursive constraint;
- Taking blanc nodes into account;

Decidability - Axioms

- More constraints on RDF;
- Also a recursive constraint;
- Taking blanc nodes into account;
- Decidability Results;

Decidability - Axioms

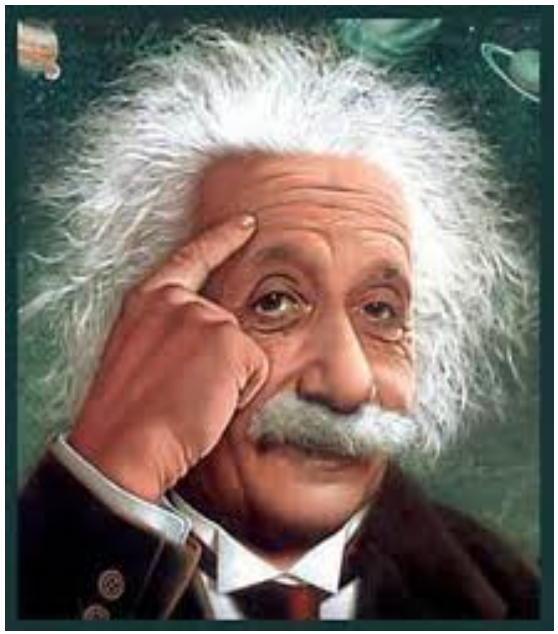
- More constraints on RDF;
- Also a recursive constraint;
- Taking blanc nodes into account;
- Decidability Results;
- Complexity Results;

Decidability - Axioms

- More constraints on RDF;
- Also a recursive constraint;
- Taking blanc nodes into account;
- Decidability Results;
- Complexity Results;
- Sound and Complete Sets of Axioms.

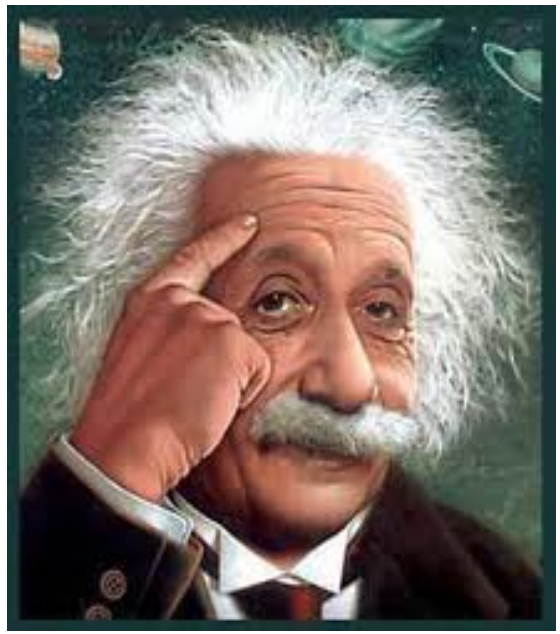
Conclusion

Conclusion



1879 - 1955

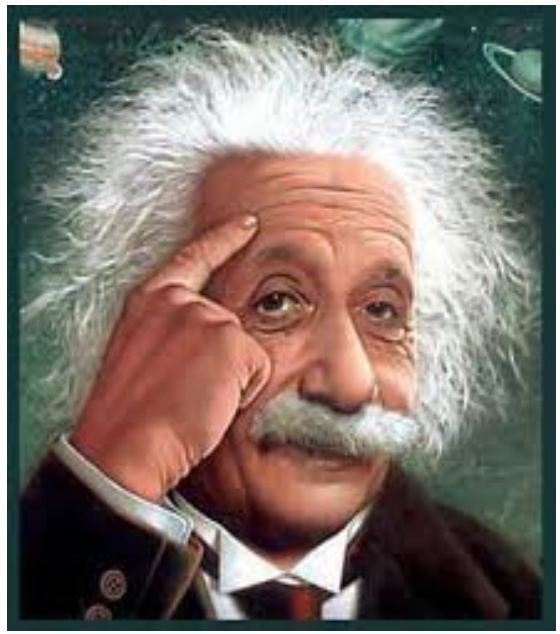
Conclusion



1879 - 1955

Theories should be as simple as possible ...

Conclusion



1879 - 1955

Theories should be as simple as possible ...

... but not simpler.

Conclusion

Video